> *A Deterministic Finite Automaton (DFA) consists of a finite set of states and a set of transitions from one state to another state on input symbols selected from an alphabet S.*

- The initial state is generally denoted by q0.
- For each input symbol there is one transition for each state.
- There are one or more “final” or “accepting” states.

The term “deterministic” says that for each input symbol, there is one and only one state to which the automaton can transit from its current state.

In case of “non-deterministic” FA the machine can make transit to zero or one or more states on the same input symbol.

> *A finite automaton is formally defined by a 5-tuple (Q, Σ, δ, q0, F),*
>
> *where,*
>
> - *Q → Finite set of states.*
> - *Σ → Finite set of input symbols.*
> - *δ → Transition function mapping Q × Σ to Q.*
> - *q0 → Initial state and q0 ∈ Q.*
> - *F → Set of final state/states. It is assumed that there may be more than one final state. F ⊆ Q.*

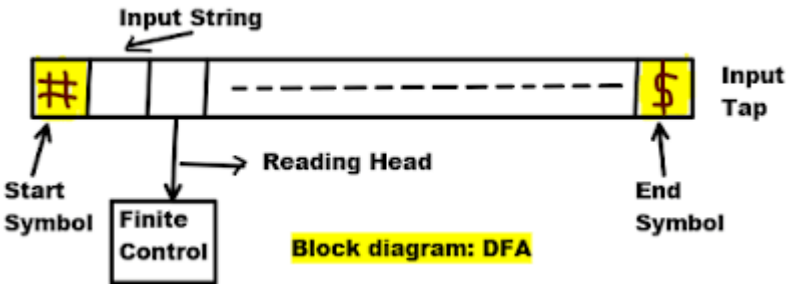
> *Reference: "Introduction to the Theory of Computation" by Michael Sipser*

An explanation of components of a DFA:

- States (Q): The states are the different possible configurations of the DFA. The DFA can be in one state at a time.
- Input symbols(Σ): The input symbols are the symbols that the DFA can read. The DFA can read one input symbol at a time.
- Transition function(δ): The transition function tells the DFA what to do when it reads an input symbol. For each state and input symbol, the transition function specifies the next state that the DFA should enter.
- Start state(q0): The start state is the state that the DFA is in when it starts processing a string.
- Accepting states(F): The accepting states are the states that the DFA can be in when it finishes processing a string. If the DFA is in an accepting state when it finishes processing a string, then the string is accepted.

The finite automation has a input tape divided into units, each containing one symbol of the input string which is built from symbols of alphabet Σ.

The start of tape is indicated by '#' and end is indicated by '$'.



- The tape may be finite or infinite.
- In case of infinite tape, the tape end marker is absent.
- The input string to be processed is stored from left to right direction.
- The reading head examines one unit at a time and can move one unit in only one direction generally, left to right.
- The current state ‘q’ and input symbol read by reading head are given as input to the finite control unit.
- The machine may remain in the same state q or transit to a new state and the reading head moves one unit ahead to read the next input symbol.
- This transition is denoted as: δ(qi, a) → qj

References:

- Introduction to the Theory of Computation” by Michael Sipser.

Related Posts:

1. Notations for DFA
2. How do a DFA Process Strings?
3. DFA solved examples

4. Definition Non Deterministic Finite Automata
5. Moore machine
6. Mealy Machine
7. Regular Expression Examples
8. Regular expression
9. Arden’s Law
10. NFA with ∈-Moves
11. NFA with ∈ to DFA Indirect Method
12. Define Mealy and Moore Machine
13. What is Trap state ?
14. Equivalent of DFA and NFA
15. Properties of transition functions
16. Mealy to Moore Machine
17. Moore to Mealy machine
18. Diiference between Mealy and Moore machine
19. Pushdown Automata
20. Remove ∈ transitions from NFA
21. TOC 1
22. Diiference between Mealy and Moore machine
23. RGPV TOC What do you understand by DFA how to represent it
24. What is Regular Expression
25. What is Regular Set in TOC
26. RGPV short note on automata
27. RGPV TOC properties of transition functions
28. RGPV TOC What is Trap state
29. DFA which accept 00 and 11 at the end of a string
30. CFL are not closed under intersection

31. NFA to DFA | RGPV TOC
32. Moore to Mealy | RGPV TOC PYQ
33. DFA accept even 0 and even 1 |RGPV TOC PYQ
34. Short note on automata | RGPV TOC PYQ
35. DFA ending with 00 start with 0 no epsilon | RGPV TOC PYQ
36. DFA ending with 101 | RGPV TOC PYQ
37. Construct DFA for a power n, n>=0 || RGPV TOC
38. Construct FA divisible by 3 | RGPV TOC PYQ
39. Construct DFA equivalent to NFA | RGPV TOC PYQ
40. RGPV Define Mealy and Moore Machine
41. RGPV TOC Short note on equivalent of DFA and NFA
42. RGPV notes Write short note on NDFA
43. Minimization of DFA
44. Construct NFA without ∈
45. CNF from S–>aAD;A->aB/bAB;B->b,D->d.
46. NDFA accepting two consecutive a’s or two consecutive b’s.
47. Regular expresion to CFG
48. Regular expression to Regular grammar
49. Grammar is ambiguous. S → aSbS|bSaS|∈
50. leftmost and rightmost derivations
51. Construct Moore machine for Mealy machine
52. RGPV TOC PYQs
53. Introduction to Automata Theory